End-to-End Contextual Speech Recognition With Word-Piece-Level Token Selection

Zhibin Wu, Yang Zou*, Jian Zhou, Min Wang, Xiaoqin Zeng
Institute of Intelligence Science and Technology, School of Computer and Information,
Hohai University, Nanjing, China
{211307040022, yzou, 211607010098, mwang, xzeng}@hhu.edu.cn

***Abstract*—The utilization of dynamic contextual information in end-to-end automatic speech recognition has been an active research topic. Generally, the popular Contextual LAS (CLAS) provides favorable all-neural solutions. Nevertheless, it cannot be extended to large bias lists without many cases of recognition errors caused by similar pronunciation or word fragment repetition. To address this limitation, this paper proposes a model called Fine-CLAS on the basis of CLAS, which exploits word-piece-level contextual knowledge and fuse it with the original phrase-level contextual knowledge to enable the contextual bias module to focus on fine-grained contextual information. First, the prefix tree constraint is presented to reduce the number of contextual phrases. Then, a strategy for word-piece-level token selection is designed to obtain the new word-piece-level embedding vector. Finally, a contextual transformation chain is constructed between the word-piece-level embedding vector key-value pairs to attain new key-value pairs. The proposed model with these techniques can reduce the word error rate (WER) by 5.37% and 2.10%, and the F1-score by 1.10% and 2.10% on the datasets test-clean and test-other of LibriSpeech, demonstrating preferable ASR and contextual bias performance.**

Keywords-dynamic contextual information; end-to-end; all-neural; word-piece-level contextual knowledge

I. INTRODUCTION

We can always feel the convenience of speech recognition technology in our lives, such as in the most commonly used smartphones, smart appliances, wearable devices, voice navigation and in-car systems [1]. In such applications, speech recognition performance can be significantly improved by incorporating information about the speaker's context into the recognition process [2]. Examples of contextual information include the status of the conversation (e.g. words such as "stop", "cancel", etc.), the location of the speaker (e.g. "restaurant", "airport", etc.) [3], personalized information about the user (e.g. contacts, song playlists, etc.) [4], and other specific nouns.

In recent years, many end-to-end automatic speech recognition (ASR) methods, such as Connectionist Temporal Classification (CTC) [5,6], Recurrent Neural Network Transducer (RNN-T) [7-12], and Attention-based Encoder-Decoder (AED) [13-18], have been widely used in life. However, the recognition of context-specific phrases in these scenarios still needs to be improved as most contextual content is scarce in the training data.

In the current work, we still consider techniques that dynamically incorporate contextual information into the recognition process. In end-to-end systems, an approach can be implemented by performing log-linear interpolation between the E2E model and the n-gram language model (LM) at each step of the beam search [14, 19-24], without adding any other neural network, which is referred to as Shallow Fusion according to the terminology in [25]. However, re-scoring using an externally trained language model independently runs counter to the benefits obtained from the joint optimization of components from sequence-to-sequence models. Thus, Golan Pundak et al. [26] proposed Contextual-LAS (CLAS), a novel all-neural mechanism that exploits contextual information (provided as a list of contextual phrases) to improve recognition performance. The technique first embeds each contextual phrase (via tokenizers, sliced into a series of word piece units) into a fixed dimensional representation, and then uses an attention mechanism to focus on the available context during decoding. In addition, a number of contextual phrases are allowed during inference. Although the full neural context approach outperforms shallow fusion, it still suffers from a problem: the performance of the model drops significantly when dealing with hundreds or even thousands of contextual phrases, which is caused by the large number of contextual phrases with similar pronunciation or partial word repetition.

To solve the problem, improvements to the CLAS model are necessary. Sun et al. [27] proposed a Tree Constrained Pointer Generator (TCPGen) component that makes full use of prefix tree selection to narrow down candidate words, enables token units at the word-piece-level, and models attention on word piece. Following this line of thought, an observation can be made that incorporating word-piece-level contextual information into the CLAS model might be a feasible way to alleviate the problems caused by word fragment repetition.

Based on this observation, this paper proposes three techniques to improve the CLAS model: prefix tree constraint, word-piece-level token selection, and contextual transformation chain construction, and the improved model is referred to as Fine-CLAS. Unlike the previous CLAS model [26] which only contextually modelled phrase-level embeddings, we propose to fuse contextual information at two different levels, phrase-level

*Corresponding author: yzou@hhu.edu.cn (Y. Zou)

DOI reference number: 10.18293/DMSVIVA23-003

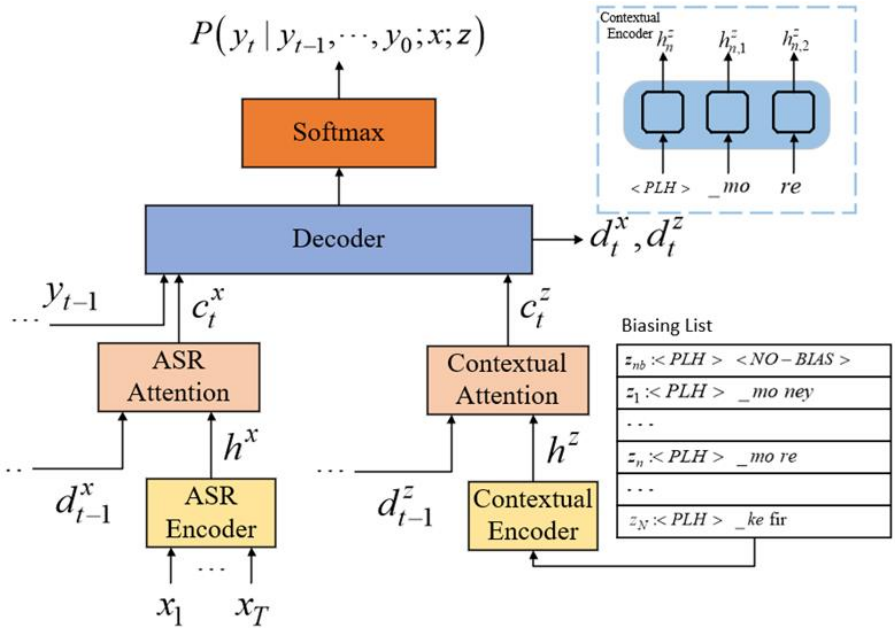

Figure 1. CLAS model: 1) The left-hand structure is the ASR of LAS and the right-hand is the context processing network; 2) The upper right-hand corner shows how the context encoder encodes a phrase and outputs its phrase embedding h_n^z and all the token embeddings $\left[h_{n,1}^z, h_{n,2}^z\right]$

embeddings and word-piece-level embeddings, in order to enable the contextual bias module to focus on fine-grained contextual information to match the ASR word-piece-level token output distribution.

The technical contributions of this paper are summarized as follows:

First, a prefix tree is constructed and combined with historical information to select whether to enable each phrase in the context list, which can reduce the number of phrases and obtain a smaller number of phrase-level biased embeddings and word-piece-level biased embeddings.

Second, a word-piece-level token selection algorithm is designed to select top-K phrases based on the weights and obtain the corresponding word-piece-level bias embeddings, which can result in a series of word-piece-level embedding information.

Third, a transformation chain between word-piece-level bias embeddings is constructed so as to obtain the transfer relationship between word-piece-level bias embeddings.

Fourth, compared to CLAS, the Fine-CLAS model constructed by incorporating the proposed techniques reduces word error rates (WER) by 5.37% and 2.10% and F1-scores (F1) by 1.10% and 2.10% on the test-clean and test-other test sets of LibriSpeech, where the list of contextual phrases consists of rare long-tail words. Furthermore, the Fine-CLAS model remains lightweight and modular, allowing for quick modifications to the contextual bias module without retraining the ASR model.

The rest of the paper is organized as follows: In Section 2 the standard AED model and the CLAS model are reviewed. In Section 3 the three techniques for improvement are described in detail. In Section 4 the experiment is described, followed by a discussion of the experimental results in Section 5. Finally, conclusions are presented in section 6.

II. BACKGROUND

A. *Attention-based Encoder-Decoder*

A standard AED contains three components: an encoder, a decoder and an attention network, as shown in the left-hand structure of Fig. 1. The encoder encodes the input $x_{1:T}$ as a sequence of high-level features h^x. In each decoding step t, the attention mechanism is utilized to combine the encoder output sequence into a single context vector c_t^x, which is used as part of the decoder input. The decoder is computed as follows.

$$d_t^x = Decoder\left(y_{t-1}, d_{t-1}^x, c_t^x\right) \quad (1)$$

where $Decoder(\bullet)$ denotes the decoder network and y_{t-1} is the embedding of the previous subword unit. The posterior distribution can be estimated using the Softmax output layer.

$$P\left(y_t \mid y_{t-1}, \cdots, y_0; x_{1:T}\right) = Soft\max\left(W^o\left[d_t^x; c_t^x\right]\right) \quad (2)$$

where $[\bullet;\bullet]$ denotes the splicing of two vectors. In the inference stage, the recognition result $y_{1:N}^*$ is calculated by performing beam search. In addition, shallow fusion [14,19-25] can be achieved by log-linear combination, as shown in the following equations.

$$y_{1:N}^* = \arg\max_{y_{1:N}} \log P\left(\mathrm{y}_{1:N} \mid x_{1:T}\right) + \lambda \log P^{LM}\left(y_{1:N}\right) \quad (3)$$

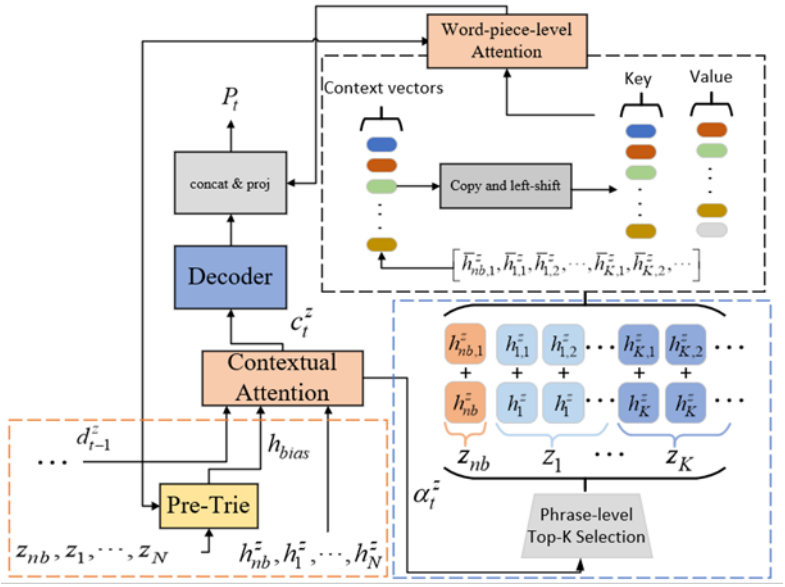

Figure 2. The structure of the Fine-CLAS model. Based on CLAS, it includes three additional modules: the prefix tree constraint, the word-piece-level token selection, and the contextual transformation chain construction, which are enclosed in the orange, blue, and black dashed boxes, respectively.

where λ is the hyperparameter controlling the relative importance of the LM output probability $P^{LM}(y_{1:N})$.

B. CLAS

CLAS models attention to contextual information, as shown in Fig. 1. The bias encoder embeds a list of biased phrases $Z=\{z_{nb}, z_1, z_2, \cdots, z_N\}$ into a set of vectors $h^z=\{h_{nb}^z, h_1^z, \cdots, h_i^z, \cdots, h_N^z\}$, where h_i^z is an embedding of z_i and $<PLH>$ is a phrase-level placeholder that represents the entire contextual phrase. Since biased phrases may be irrelevant to the current discourse, we introduce the phrase-level unbiased option z_{nb}. The embedding h_i^z is created by feeding a sequence of subword embeddings in z_i (i.e. the same lexical elements or chunk units used by the decoder) to the biased encoder and representing the whole phrase using the first state output of the LSTM. Attention modelling is then performed at h^z, using the decoder state d^t to compute the auxiliary context vector c_t^z. This context vector summarizes z at time step t and is calculated as shown below.

$$u_{it}^z = v^{z^T} \tanh\left(W_h^z h_i^z + W_d^z d_t + b_a^z\right) \quad (4)$$

$$a_t^z = soft\max\left(u_t^z\right) \quad (5)$$

$$c_t^z = \sum_{i=0}^{N} a_{it}^z h_i^z \quad (6)$$

Next, the context vector c_t^x, obtained by combining the ASR attention, yields the LAS context vector $c_t = \left[c_t^x; c_t^z\right]$ for the input decoder. It is worth noting that, given the audio and the previous output, CLAS can obtain the weights of the bias phrases that are of interest during the current decoding process, as follows.

$$a_t^z = P(z_t \mid d_t) = P(z_t \mid x; y_{<t}) \quad (7)$$

We refer to a_t^z as bias-attention-probability.

III. METHODS

The Fine-CLAS model is established on the CLAS model by augmenting three additional models that correspond to three approaches, as shown in Fig. 2. First, the prefix tree constraint is introduced to reduce the number of contextual phrases. Then word-piece-level token selection is performed to obtain the new word-piece-level embedding vector. Finally, a contextual transformation chain construction is executed between the word-piece-level embedding vector key-value pairs (K and V) to obtain new key-value pairs, which are used in the computation of the word-piece-level attention mechanism to obtain the final word-piece-level context vector.

A. Prefix Tree Constraint

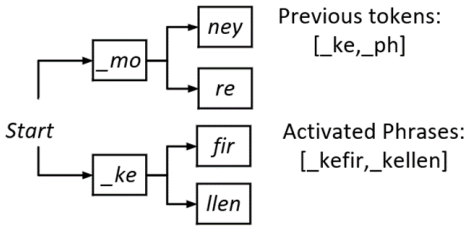

Figure 3. An example of prefix tree search.

In this subsection, we propose a trie-based bias module that encodes the bias list into a prefix tree at the word-piece-level, as shown in Fig. 3. Given the previously output word fragment tokens as queries, a certain history interval is selected and input to the bias module to find the phrases corresponding to the prefixes, returning a binary vector $h_{bias} = [a_0, a_1, \cdots, a_N] \in \{0,1\}$, with N being the number of phrases. $a_n = 0$ means the phrase is not activated and not relevant to the sentence; $a_n = 1$ means the phrase is activated and relevant to the sentence. h_{bias} is computed to filter relevant phrases and will only be used for phrase-level attention in the inference stage, as shown in the orange dashed box in Fig. 2.

B. Word-piece-level Token Selection

In this subsection, we propose a word-piece-level token selection technique. It introduces word-piece-level context vectors that are spliced and mapped to the decoder's output, thus matching the token units of ASR with word fragments as the output distribution and reducing the uncertainty of token prediction, as shown in the blue dashed box in Fig. 2.

First, the token-level acoustic embedding vector d_{t-1}^z for the current time step t is modeled with a series of phrase-level bias embedding vectors $h^z = [h_{nb}^z, h_1^z, \cdots, h_N^z]$ for phrase-level attention, resulting in phrase-level context weights $a_t^z = [a_{t,nb}^z, a_{t,1}^z, \cdots, a_{t,N}^z]$. Then the average attention weight $\breve{a}_t^z = [\breve{a}_{t,nb}^z, \breve{a}_{t,1}^z, \cdots, \breve{a}_{t,N}^z]$ is calculated based on the global (time step t with all previous attention) or local (time step t with the attention of the previous finite time step). The size of the list of context-biased phrases can be hundreds or thousands, which is not small even after prefix tree filtering. If we directly use the word-piece-level embedding vector for each phrase, the corresponding list of word-piece-level embedding vectors will become very large. So we select top-K attention weights from $\breve{a}_t^z$, and then get the corresponding contextual bias phrases according to the index of the selected weights to achieve the reduction from N to K. For each bias phrase selected, the first state output h_k^z of the encoder representing the phrase-level embedding vector is respectively added to the subsequent state output $h_{k,i}^z$ of the encoder representing the word-piece-level embedding vector, and we get a series of word-piece-level embedding vectors corresponding $\bar{h}_{k,i}^z$, which results in a list of all word-piece-level embedding vectors $K = V = [\bar{h}_{nb,1}^z, \bar{h}_{1,1}^z, \bar{h}_{1,2}^z \cdots, \bar{h}_{K,1}^z, \bar{h}_{K,2}^z, \cdots]$. The specific formula is as follows.

$$[z_{nb}, z_1, \cdots, z_K] = PhraseTopKSelection\left(Z, [\breve{a}_{t,nb}^z, \breve{a}_{t,1}^z, \cdots, \breve{a}_{t,N}^z]\right) \quad (8)$$

$$[h_k^z, h_{k,1}^z, h_{k,2}^z, \cdots] = ContextualEnc(z_k) \quad (9)$$

$$[h_{nb}^z, h_{nb,1}^z] = ContextualEnc(z_{nb}) \quad (10)$$

$$\bar{h}_{k,i}^z = h_k^z + h_{k,i}^z \quad (11)$$

$$\bar{h}_{nb,1}^z = h_{nb}^z + h_{nb,1}^z \quad (12)$$

$$K = V = [\bar{h}_{nb,1}^z, \bar{h}_{1,1}^z, \bar{h}_{1,2}^z, \cdots, \bar{h}_{K,1}^z, \bar{h}_{K,2}^z, \cdots] \quad (13)$$

C. Contextual Transformation Chain Construction

Although in the word-piece-level token selection technique, word-piece-level contexts are constructed for use in the decoding step to achieve fine-grained local bias, the probability of transfer between word fragment tokens are not explicitly modelled. Modelling this transfer may be helpful when the context is personalized entity names and proper names that are rare or invisible during training, as it allows us to recover the expected next token by using the preceding subsequence. We therefore introduce a more fine-grained biasing technique that operates at the word-piece-level, following word-piece-level token selection, as shown in the black dashed box in Fig. 2.

Specifically, we construct an associative memory to store and retrieve the associated bias context. As shown in Fig. 2, the memory stores association transfers between word-piece-level subsequences of the same phrase. In the associative memory, the key of each word-piece-level token in each phrase is mapped to the value of the next word-piece-level token (left shift). The original formula for the key-value pair selected by the word-piece-level token is as follows.

$$k_l = v_l = \bar{h}_{k,i}^z \quad (14)$$

Accordingly, the memory entries of the key-value pair (k_l, v_l) constructed after the contextual transformation chain are two consecutive word-piece-level embedding vectors $\bar{h}_{k,i}^z$ and $\bar{h}_{k,i+1}^z$, as follows.

$$(k_l, v_l) = (\bar{h}_{k,i}^z, \bar{h}_{k,i+1}^z) \quad (15)$$

IV. Experiments

A. Datasets and Metrics

Our experiments are conducted on the dataset Librispeech. The dataset is collected from an audiobook website, and speech recognition is done once for each sentence. The acoustic model from the WSJ example is adopted as the recognition model, a binary grammar is utilized as the language, and the input dataset for the language model is the e-book text corresponding to the speech data. From the clean data, 20 males and 20 females are randomly selected as the development set (dev-clean), the remaining speakers are selected as a test set of the same size (test- clean), and the rest as the training set. The training set is 100 hours (train-clean-100). In the other data, the WERs are sorted from lowest to highest, and the test set is randomly selected near the third quartile (test-other). As LibriSpeech's test set lacks a bias list, we construct a bias list by collecting words other than the 20,000 most common words in the training data from the reference of the test set and discarding short words of

less than 5 letters. Finally, the simulated bias lists for test-clean and test-other consists of around 1,000 phrases.

Firstly, a set of evaluation metrics is introduced that tracks three different aspects of ASR, (1) WER: overall word error rate assessed for all words, (2) CER: overall character error rate assessed for all words, (3) U-WER: unbiased word error rate assessed for words not in the bias list. Secondly, contextual bias is measured using the precision (P), recall (R) and F1-score (F1) of the biased phrases. In summary, we use six evaluation metrics to measure the performance of Fine-CLAS.

B. Configurations

The model evaluated in this paper is trained on an A40 graphics card with 48G of video memory and a batch size of 8. To improve the performance of the model, the data enhancement method of SpecAugment is used. The input features are a 40-dimensional log-mel filter bank with a sampling rate of 16000Hz, extracted from a window of length 25ms, length of the hop of the sliding window is 10ms, and its output vocabulary is a 1000-word block generated via BPE.

The ASR encoder is composed of a convolutional module, a cyclic module and a fully connected module. The convolutional module consists of two 3x3 convolutional layers with 128 and 256 nodes, the cyclic module includes four bi-directional LSTM layers with 1024 nodes each and the fully connected module comprises two fully connected layers with 512 nodes. The ASR encoder attention is computed in 1024 dimensions using a content-based attention mechanism. The decoder contains 1 GRU with 1024 nodes. The context encoder involves 1 bi-directional LSTM layer with 128 nodes, and the phrase-level attention and word-piece-level attention have the same structure as the ASR attention.

The model has a total of 177.4M trainable parameters and our model is implemented using Pytorch and Speechbrain.

In order to exercise the "no bias" option, we use the same settings as in [26]. In all experiments, we set $P_{keep} = 0.5$ to improve robustness to the "no bias" case, and set $N_{phrase} = 1$ and $N_{order} = 4$. This results in an expected size of 5 for the bias list (half the batch size, plus one "no bias" option). In addition, the phrase selection has K of 5. In inference, we adopt a beam size of 10 for the search.

V. RESULTS

A. Evaluation Results for ASR

TABLE I. ASR test results on test-clean and test-other

Model	test-clean			test-other		
	WER	CER	U-WER	WER	CER	U-WER
AED	21.79	10.98	19.90	45.65	26.35	41.30
CLAS	22.97	16.37	19.90	38.18	22.63	33.90
Fine-CLAS	**17.60**	**10.63**	**16.00**	**36.08**	**20.95**	**32.70**

We use the simulated bias list to validate the improvements to the model, and evaluate the performance of the model ASR on three metrics. As shown in Table I, AED achieves a word error rate of 21.79% on test-clean and 45.65% on test-other, which is the result of testing without additional language model. Compared with AED, the improved CLAS model shows an increase in WER and CER on test-clean and a noticeable decrease in WER and CER on test-other, indicating that the ASR performance of the CLAS model is quite good. Compared with CLAS, our improved Fine-CLAS model decreases WER by another 5.37% and 2.10% on test-clean and test-other, respectively, and achieves noticeable improvements in the other two metrics. This indicates that the ASR performance of our model is preferable.

B. Evaluation Results for Contextual Biasing

TABLE II. Contextual bias test results on test-clean and test-other

Model	test-clean			test-other		
	F	R	F1	F	R	F1
AED	**97.10**	37.80	54.40	82.40	15.60	26.20
CLAS	92.90	59.80	72.80	88.80	29.40	44.10
Fine-CLAS	96.00	**66.10**	**73.90**	**95.50**	**30.40**	**46.20**

To test the effectiveness of the proposed model's contextual bias, we use three evaluation metrics, as shown in Table Ⅱ. First, the CLAS model achieves better performance than the AED, especially in the F1-score metric, which is improved by almost 20%, indicating that the CLAS model noticeably improves the contextual bias effect. Compared to CLAS, our Fine-CLAS model achieves a slight improvement with another 1.10% and 2.10% improvement in F1-score on test-clean and test-other, respectively. This indicates that our model improves both the performance of the ASR model and the effect of contextual bias.

VI. CONCLUSION

In this work, we propose the Fine-CLAS model that promotes end-to-end contextual speech recognition through three techniques: prefix tree constraint, word-piece-level token selection, and contextual transformation chain construction. The improved model can mitigate confusion caused by similar pronunciations or word fragment repetition. The experimental results of several evaluation metrics on the dataset LibriSpeech clearly show that these proposed techniques improve the performance of the original context-biased approach and make the Fine-CLAS model more capable of handling a large number of contextual phrases. In the future work, we shall attempt to further expand the context bias list and explore even better methods for dealing with contextual issues. In addition, we shall attempt to combine it with ChatGPT, an AI chatbot, to explore multimodal contextualization from speech to text.

REFERENCES

[1] I. McGraw et al., “Personalized speech recognition on mobile devices,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5955-5959, 2016.

[2] P. Aleksic, M. Ghodsi, A. Michaely, et al., “Bringing Contextual Information to Google Speech Recognition,” Proc. Interspeech, pp. 468-472, 2015.

[3] J. Scheiner, I. Williams and P. Aleksic, “Voice search language model adaptation using contextual information,” IEEE Spoken Language Technology Workshop, pp. 253-257, 2016.

[4] P. Aleksic, C. Allauzen, D. Elson, A. Kracun, D. M. Casado and P. J. Moreno, “Improved recognition of contact names in voice commands,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5172-5175, 2015.

[5] A. Graves, S. Fernández, F. Gomez, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” Association for Computing Machinery, pp. 369-376, 2006.

[6] A. Graves, N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” International Conference on Machine Learning, pp. 1764-1772, 2014.

[7] A. Graves, “Sequence Transduction with Recurrent Neural Networks,” Computer Science, arXiv preprint arXiv:1211.3711, 2012.

[8] A. Graves, A.R. Mohamed, G. Hinton, “Speech recognition with deep recurrent neural networks,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645-6649, 2013.

[9] R. Prabhavalkar, K. Rao, T. Sainath, B. Li, L. Johnson & N. Jaitly, “A comparison of sequence-to-sequence models for speech recognition,” Proc. Interspeech, pp. 939-943, 2017.

[10] E. Battenberg, J. Chen, R. Child, et al., “Exploring neural transducers for end-to-end speech recognition,” IEEE Automatic Speech Recognition and Understanding, pp. 206-213, 2017.

[11] J. Li, R. Zhao, H. Hu and Y. Gong, “Improving RNN Transducer Modeling for End-to-End Speech Recognition,” IEEE Automatic Speech Recognition and Understanding, pp. 114-121, 2019.

[12] Y. He et al., “Streaming End-to-end Speech Recognition for Mobile Devices,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6381-6385, 2019.

[13] J. Chorowski, D. Bahdanau, D. Serdyuk, et al., “Attention-Based Models for Speech Recognition, ” Neural Information Processing Systems, pp. 577-585, 2015.

[14] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4945-4949, 2016.

[15] L. Dong, S. Xu and B. Xu, “Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 5884-5888, 2018.

[16] C.C. Chiu, T.N. Sainath, Y. Wu, et al., “State-of-the-art Speech Recognition With Sequence-to-Sequence Models,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4774-4778, 2018.

[17] A. Zeyer, K. Irie, R. Schlüter, et al., “Improved training of end-to-end attention models for speech recognition,” Proc. Interspeech, pp. 7-11, 2018.

[18] W. Chan, N. Jaitly, Q. Le and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, ” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4960-4964, 2016.

[19] I. Williams, A. Kannan, P. Aleksic, D. Rybach, T. Sainath, “Contextual speech recognition in end-to-end neural network systems using beam search,” Proc. Interspeech, pp. 2227-2231, 2018.

[20] Z. Chen, M. Jain, Y. Wang, M. L. Seltzer and C. Fuegen, “End-to-end Contextual Speech Recognition Using Class Language Models and a Token Passing Decoder,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6186-6190, 2019.

[21] D. Zhao, T. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li & R. Pang, “Shallow-fusion end-to-end contextual biasing,” Proc. Interspeech, pp. 1418-1422, 2019.

[22] R. Huang, O. Abdel-Hamid, X. Li, et al., “Class LM and word mapping for contextual biasing in End-to-End ASR,” Proc. Interspeech, pp. 4348-4351, 2020.

[23] Y.M. Kang, Y. Zhou, “Fast and Robust Unsupervised Contextual Biasing for Speech Recognition,” arXiv preprint arXiv:2005.01677, 2020.

[24] C. Liu, D.R. Liu, F. Zhang, et al., “Contextualizing ASR Lattice Rescoring with Hybrid Pointer Network Language Model,” Proc. Interspeech, pp. 3650-3654, 2020.

[25] C. Gulcehre, O. Firat, K. Xu, et al., “On Using Monolingual Corpora in Neural Machine Translation,” arXiv preprint arXiv:1503.03535, 2015.

[26] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan and D. Zhao, “Deep Context: End-to-end Contextual Speech Recognition,” IEEE Spoken Language Technology Workshop, pp. 418-425, 2018.

[27] G. Sun, C. Zhang and P. C. Woodland, “Tree-Constrained Pointer Generator for End-to-End Contextual Speech Recognition,” IEEE Automatic Speech Recognition and Understanding Workshop, pp. 780-787, 2021.